

Globus User Guide (for LC)

Table of Contents

Preface	3
Introduction	4
Using Globus at LC	5
Submitting Jobs to Run at LC	5
Submitting Jobs at LC to Run Elsewhere	9
GLOBUSRUN	10
GLOBUSRUN Execute Line	10
GLOBUSRUN Options	11
GLOBUSRUN Examples	13
Resource Specification Language	15
RSL Syntax Summary	15
RSL Special Characters	18
RSL Attribute Dictionary	19
ASCI Grid Monitoring Services	22
Disclaimer	25
Keyword Index	26
Alphabetical List of Keywords	27
Date and Revisions	28

Preface

- Scope:** Globus is a multi-agency software project (and hence the toolkit of programs developed by that project) to enable easy, effective sharing of widely distributed high-performance computers (and their supporting resources). This Globus User Guide (for LC) supplements the broad, general descriptions and instructions on the Globus web site (www.globus.org (URL: <http://www.globus.org>)) by emphasizing just the features of Globus most relevant to preparing jobs for and remotely running jobs on LC production machines (especially the ASCI White IBM SP). It covers the job-control steps needed (with LC local modifications spelled out), the portion of GLOBUSRUN options most relevant to guiding batch jobs at LC, the subset of Resource Specification Language (RSL) features most helpful for characterizing these jobs, and how to monitor the ASCI Grid with Globus tools.
- Availability:** When the programs described here are limited by machine, those limits are included in their explanation. Otherwise, they run under any LC UNIX system.
- Consultant:** For help contact the LC customer service and support hotline at 925-422-4531 (open e-mail: lc-hotline@llnl.gov, SCF e-mail: lc-hotline@pop.llnl.gov).
- Printing:** The print file for this document can be found at:
- on the OCF: <http://www.llnl.gov/LCdocs/globus/globus.pdf>
on the SCF: https://lc.llnl.gov/LCdocs/globus/globus_scf.pdf

Introduction

Globus is a multi-agency software project (and hence the toolkit of programs developed by that project) to enable easy, effective sharing of widely distributed high-performance computers (and their supporting resources). Offsite, nonLLNL ASCI collaborators can submit their collaborative MPI projects to run on the ASCI White IBM SP machine (when available) by using Globus (rather than PSUB) as a remote front end to LLNL's local batch system. The Globus project is a team effort to promote distributed supercomputing by Argonne National Laboratory, NCSA, NASA, the University of Chicago, and the University of Southern California.

Globus takes a layered approach to managing remote (batch) jobs and to assigning resources to them. The elaborate background details (and extensive jargon) are discussed in many files connected to the Globus web site (www.globus.org) (URL: <http://www.globus.org>). This simplified diagram, however, summarizes the key terms and relationships most relevant to LC users of Globus:

The Globus "Computational Grid" Architecture

Layer:	Examples:
Applications	[your script or code]
Tools and toolkits	grid-proxy-init, globusrun, Resource Specification Language RSL
Underlying services	GRAM = Globus Resource Allocation Manager GIS = Grid Information Service (formerly Metacomputing Directory Service MDS)
Grid fabric of resources	computers, such as ASCI White; job managers, such as LC's DPCS and (under it) IBM's LoadLeveler

Globus lets you authenticate yourself through its services, then use its tools on your local computer to submit and manage (batch) jobs that run remotely on LC machines.

This Globus User Guide (for LC) emphasizes just the features of Globus most relevant to preparing jobs for and remotely running jobs on LC production machines (especially the ASCI White IBM SP). The sections below cover:

- the job-control steps needed (with LC local modifications spelled out),
- the portion of GLOBUSRUN options most relevant to guiding batch jobs at LC,
- the subset of Resource Specification Language (RSL) features most helpful for characterizing these jobs, and
- how to monitor the "ASCI Grid" using a Globus-based web site (and the related but independent DISCOM NETMON web site).

Using Globus at LC

Submitting Jobs to Run at LC

This section tells LC collaborators elsewhere how to prepare and submit (batch) jobs to run on LC production machines (primarily ASCI White) by using Globus software for job control. (LC users on LC machines should simply use DPCS (URL: <http://www.llnl.gov/LCdocs/dpcs>) and the local PSUB utility, as explained in EZJOBCONTROL (URL: <http://www.llnl.gov/LCdocs/ezjob>).)

You need to perform the first four steps below only ONCE, to prepare generally for using Globus to submit jobs. You need to perform all steps after the first four EVERY TIME you run a job (although related jobs often share some preparation from one to another, of course).

1. Request account(s)
 2. Select "home" computer
 3. Enable Globus software tools
 4. Get Globus certification
- general
-----specific
5. Get a Globus proxy
 6. Describe your job's resource needs (RSL)
 7. Submit your job (GLOBUSRUN)

1. REQUEST ACCOUNTS(S).

Before you can submit any job to a machine (such as ASCI White), you need an authorized account on that machine. For general background on LC user authorization, policies, and application forms, consult the EZACCESS (URL: <http://www.llnl.gov/LCdocs/ezaccess>) Basic Guide. For personal help applying for an LC account (or checking on your account status), contact the LC Hotline at 925-422-4531 (open e-mail: lc-hotline@llnl.gov).

2. SELECT HOME COMPUTER.

Successful interactions with remote Globus software (such as when you run jobs remotely and track their status) requires that you choose and maintain one "home" computer. This machine must have Globus client software installed already. When you apply for a Globus certificate starting from one machine (see step four below), the certificate authority will henceforth regard that as your home machine for all future Globus interactions.

3. ENABLE GLOBUS SOFTWARE TOOLS.

On your Globus "home" machine, you need to

- (a) remove any old(er) Globus directories from your search path,
- (b) set the GLOBUS_INSTALL_PATH environment variable to where Globus is installed on your system, and
- (c) add the (current) Globus tools directory to your search path.

Check with your system administrator, since the exact steps depend on where Globus resides, what shell you are running, and what default settings are in place before you start. You may want to add lines to your run control files to facilitate these settings, such as (for .cshrc):

```
if ( $?GLOBUS_INSTALL_PATH ) then
    if ( -f ${GLOBUS_INSTALL_PATH}/etc/globus-user-setup.csh ) then
        source ${GLOBUS_INSTALL_PATH}/etc/globus-user-setup.csh
    endif
endif
```

or (for .profile):

```
if [ "${GLOBUS_INSTALL_PATH}" ]; then
    if [ -f ${GLOBUS_INSTALL_PATH}/etc/globus-user-setup.sh ]; then
        . ${GLOBUS_INSTALL_PATH}/etc/globus-user-setup.sh
    fi
fi
```

4. GET GLOBUS CERTIFICATION.

A Globus "certificate" allows you to authenticate yourself once, then use many Globus resources (or the same resource many times) without further approvals. Getting a certificate involves many steps, usually spread over several days:

(a) Log in to your chosen home computer, run the Globus tool

```
grid-cert-request
```

and respond to its request for a passphrase with a string at least 8 characters long, mixed case, with some letters and numerals. Save the passphrase; you will need it for every Globus run.

(b) Running GRID-CERT-REQUEST

[i] creates a dot-globus (.globus) directory in your home directory,
[ii] creates a file called usercert_request.pem within the .globus directory, and
[iii] in its output tells you how (to use CAT and MAIL) to e-mail this file to your "Globus Certificate Authority," which for users of LC resources is located at Sandia National Laboratory (current details are in the program output).

(c) E-mail your certificate-request file as instructed (obviously, your home computer must be able to send and receive e-mail for this step to work). You should get a response (which will become the file usercert.pem) within two days; if not, send a query (but NOT another certificate request) to the certificate authority address.

(d) Save the e-mail response to your certificate request into the file

```
~/ .globus/usercert.pem
```

Overwrite any former usercert.pem file you may have had. Then change the permissions on the .pem files in .globus using CHMOD:

```
chomod 400 userkey.pem
chmod 444 usercert.pem
```

Your "signed public key" is now available for authentication whenever you run Globus tools.

(e) Open usercert.pem (your certificate response), and find your Globus "subject name" (the long quoted string illustrated here starting with /C=):

```
Enclosed is your Globus Certificate for:
"/C=US/O=Globus/O=NASA Ames Reserach Center/OU=Numerical
Aerospace Simulation/CN=Jane Smith"
```

Then e-mail a copy of this string to the (Globus) administrator for each Globus resource you plan to use. For LC (the ASCI White machine), mail this to either Moe Jette (jette1@llnl.gov) or Keith Fitzgerald (fitzgerald2@llnl.gov). Their response will confirm that you have been added to the grid-mapfile on the target resource, and hence that your certification process is complete.

(f) You can execute

```
globus-setup-test
```

on your home computer to confirm your certification to run jobs (an "authorization failure" response indicates that you are not yet in the grid-mapfile for LC).

5. GET GLOBUS PROXY.

Each time you want to submit a Globus job to run on an LC machine, you must get a new Globus proxy (credential) with a lifetime long enough to cover the job's entire wall-clock run time. On your Globus-certified "home" computer, run

```
grid-proxy-init -hours hhh
```

This prompts for your Globus passphrase (selected in step 4(a)) and issues a proxy for the next *hhh* hours, which must cover your job's total execution (default is 12 hours). To destroy a no longer needed proxy before it expires, type

```
grid-proxy-destroy
```

and to query the current status of your last Globus proxy, type

```
grid-proxy-info -all
```

6. DESCRIBE RESOURCE NEEDS (RSL).

Create a file to describe your job's resource needs (and to make repeat runs easy and consistent). This file is not an executable script but rather an inventory of job requirements written in the Globus Resource Specification Language (RSL). A sample (portion of an) RSL file is:

```
& (executable = /g/g0/xyz/globus/project3)
  (maxCpuTime = 500)
  (count = 8)
...
```

Here PROJECT3 can be simply a binary program, but usually is an executable shell script that includes all the typical job-control steps (such as setting up the environment, retrieving files from storage, and

managing output). See EZJOBCONTROL (URL: <http://www.llnl.gov/LCdocs/ezjob>) for advice on preparing such a job-control script suitable for batch jobs on LC machines. See the RSL section (page 15) below for an LC-focused summary of RSL syntax, a list of special characters, and a dictionary of supported job attributes, including examples and defaults. WARNING: in RSL files you must use full pathnames for all files and fully qualified names for all machines (e.g., rx12.llnl.gov, not just rx12).

7. SUBMIT YOUR JOB (GLOBUSRUN).

To execute your job (usually a script that invokes a whole series of coordinated utility and application programs) remotely on an LC machine (such as ASCI White), log on to your Globus "home" computer and type

```
globusrun -b -f fname -r host/jobmanager-dpcs
```

where

- b specifies a batch job to run on batch-available nodes,
- f specifies the RSL file *fname* that describes your job's resource needs and that identifies its executable script (as you created in step 6), and
- r specifies your target Globus "resource" (a fully qualified *host* name, such as rx12.llnl.gov, concatenated with the Globus identifier for the "service" your job will use on that host, always "dpcs" for batch jobs.

For more on the GLOBUSRUN execute line, options, and usage examples, see the GLOBUSRUN section (page 10) below. NOTE: at LC, if a DPCS (job-control) attribute is set several times, the last setting replaces all others. This entails that if you happen to specify resource needs in both your RSL file and (with embedded PSUB commands) in the job's executable script too, those in the script will override those in the RSL file. Keep this in mind if you need to change the job's resources.

Submitting Jobs at LC to Run Elsewhere

This service is not yet available, and, when implemented, may not involve running Globus tools (but rather an extended version of PSUB).

GLOBUSRUN

GLOBUSRUN Execute Line

The GLOBUSRUN utility submits a job to run remotely on a Globus-managed computer. For LC users, these are invariably batch (not interactive) jobs whose complex resource needs you have described in an RSL specification file. (page 15) Two similar tools, called GLOBUS-JOB-RUN and GLOBUS-JOB-SUBMIT, are by contrast intended for remote execution of simple (no RSL) jobs executing interactively rather than in batch.

AFTER following all the preparatory steps in the previous section (page 5), you can submit a remote Globus job by logging on to your Globus-certified "home" computer and typing

```
globusrun -b -f fname -r host/jobmanager-dpcs
```

or, more generally,

```
globusrun options
```

where, however, most *options* are irrelevant to LC jobs or are even disabled by the -b[atch] option. The next section (page 11) explains the GLOBUSRUN options, with emphasis on those three (-b, -f, -r) most relevant for LC jobs.

When you execute GLOBUSRUN, the program contacts "gatekeeper" software on the remote target machine (specified by -r, in this case ASCI White) and performs mutual (user/machine) authentication. Then the gatekeeper hands your job off to the local "job manager service" (in this case, LC's Distributed Production Control System or DPCS). DPCS schedules your incoming Globus job along with other batch jobs based on your job's resource needs and the other batch jobs currently competing for available computing resources.

You can use the -l[ist] option of GLOBUSRUN or a separate utility called GLOBUS-JOB-STATUS (without options) to check if your job has started to run or has completed.

GLOBUSRUN Options

Most GLOBUSRUN options are not relevant for LC users running batch jobs remotely on ASCI White. Therefore, this section describes the crucial options first, then other generally applicable options (for job status and help), then finally and briefly the irrelevant options (unneeded for LC or incompatible with the `-b[atch]` option).

Crucial GLOBUSRUN Options:

- `-b[atch]` submits the job to the remote machine's batch queue and disconnects from it once submitted. `-b` implies `-q[uiet]` and is incompatible with `-interactive`, `-server`, `-output`, and `-write-allow`.
- `-f[ile] fname` reads the job's resource specifications from the file *fname* (a plain text file written in the Globus Resource Specification Language RSL) rather than from the execute line. The specifications for most large batch jobs that run on LC machines are fairly complex, and using an RSL file allows easy debugging, between-run comparison, and reliable reuse.
- `-r[esource] hostname/servicename`
 - specifies the job's remote target machine and execution "service," where
 - hostname* is the fully qualified name of the target machine. At LC, this is a login node's full name (such as `rx12.llnl.gov`). Globus checks with its Grid Information Service (GIS, often called Metacomputing Directory Service MDS in older Globus documentation) to verify and resolve the host name.
 - servicename* is the string
 - `jobmanager-dpcs`
 - for all batch jobs run on LC machines (it would be `jobmanager-fork` if remote interactive jobs were allowed).

Job Status Options:

- `-l[ist]` reports all previously started and not-yet-finished jobs. For each job, GLOBUSRUN lists the *jobid* (needed as input for `-status` and `-kill`) and the job's RSL string.
- `-status jobid` reports the current state of the job specified by *jobid*, where the possible states are PENDING, ACTIVE, DONE, SUSPENDED, and FAILED.
- `-k[ill] jobid` kills the job specified by *jobid*.

Help Options:

- help displays the GLOBUSRUN help message, a usage synopsis.
- usage displays the GLOBUSRUN help message, a usage synopsis.
- version displays the current GLOBUSRUN version number.

Debugging Options:

- p[arse] parses your RSL (resource specification) string or file and reports success or detected (syntax) failures. This is a test; no job is submitted to any actual Globus machine.
- a[uthenticate-only] pings your specified Globus resource manager to detect authentication failures or proxy expiration. This is a test; no job is submitted to any actual Globus machine.
- d[ryrun] adds the clause dryrun=yes to your job's RSL and sends it to the target machine to detect and report nonsyntactic RSL errors (such as bad directories or bad executables). This is a test; your job is submitted to but does not actually start to run on your target Globus machine.

Control and Output Options (NOT relevant for LC jobs):

- i[nteractive] starts an interactive Globus job on multiple machines.
- mdshost *host*
- mdsport *port*
- mdsbasedn *basedn*
- mdstimeout *secs*
overrides the default settings of the environment variables GRID_INFO_HOST, GRID_INFO_PORT, GRID_INFO_BASEDN, and GRID_INFO_TIMEOUT respectively (to help the Grid Information Service/MDS resolve resource descriptions).
- n[o-interrupt] disables your ability to interrupt interactive Globus jobs by typing CTRL-C.
- o[utput] (incompatible with -b, implies -q) limits output to the standard output and standard error streams.
- q[uiet] (entailed by -b) omits all diagnostic messages in GLOBUSRUN output.
- s[erver] (incompatible with -b, implies -q and -o) same as -output, and also provides read-only service to the local file system.
- w[rite-allow] (incompatible with -b, implies -q, -o, and -s) same as -s but also allows writing to the local file system.

GLOBUSRUN Examples

These examples use the Globus Resource Specification Language RSL. See the [RSL section](#) (page 15) below for an LC-focused summary of RSL features and a dictionary of supported job attributes and their defaults.

[1]

GOAL: On a remote Globus-certified host, to submit a batch job to an LLNL/LC machine, where a local file summarizes the job's resource needs.

STRATEGY: (1) See the [above section](#) (page 5) on "Submitting Jobs to Run at LC" for the many prerequisite steps you need BEFORE you use GLOBUSRUN.
(2) Make a file in your local directory that characterizes your job using RSL. For example (jobspec),

```
&
(executable = /g/g0/xyz/globus/myjob)
(maxTime = 500)
(hostCount = 2)
(count = 8)
(environment = (MP_EUILIB us))
(queue = pbatch)
(stdout = /g/g0/xyz/globus/myjob.o)
(stderr = /g/g0/xyz/globus/myjob.e)
```

(3) Execute GLOBUSRUN with the -b[atch] option, with -r to specify the fully qualified name of the target node and "service" (here rx12.llnl.gov and DPCS), and with -f to specify the RSL file (here, jobspec) that describes the job.

```
globusrun -b -r rx12.llnl.gov/jobmanager-dpcs -f jobspec
```

[2]

GOAL: On a remote Globus-certified host, to submit a batch job to an LLNL/LC machine, where you describe the job's resource needs by tying RSL expressions directly on the GLOBUSRUN execute line.

STRATEGY: (1) See the [above section](#) (page 5) on "Submitting Jobs to Run at LC" for the many prerequisite steps you need BEFORE you use GLOBUSRUN.
(2) Plan your job's resource needs, using the attributes that would go into an RSL file.
(3) Execute GLOBUSRUN with the -b[atch] option, with -r to specify the fully qualified name of the target node and "service" (here rx12.llnl.gov and DPCS), and, instead of -f, type directly into the execute line the RSL clauses that describe the job (note the surrounding quotes and the slash (\) line-continuation character).

```
globusrun -b -r rx12.llnl.gov/jobmanager-dpcs "& \  
  (executable = /g/g0/xyz/globus/myjob) \  
  (maxTime = 500) \  
  (hostCount = 2) \  
  (count = 8) \  
  (environment = (MP_EUILIB us)) \  
  (queue = pbatch) \  
  (stdout = /g/g0/xyz/globus/myjob.o) \  
  (stderr = /g/g0/xyz/globus/myjob.e) "
```

[3]

GOAL: On a remote Globus-certified host, to discover the status of your currently submitted Globus jobs.

STRATEGY: (1) See the [above section](#) (page 5) on "Submitting Jobs to Run at LC" for the many prerequisite steps you need BEFORE you use GLOBUSRUN.
(2) Use GLOBUSRUN's -l[ist] option to report for each job started and not yet completed the Globus job ID and the job's RSL string.
(3) Pipe the GLOBUSRUN output into a filter (such as AWK, shown here) to pick out only relevant jobs (such as those called project3).

```
globusrun -l | awk '/project3/ {print $1}'
```

Resource Specification Language

RSL Syntax Summary

The Globus Resource Specification Language (RSL) provides a neutral way to describe the resource needs of submitted jobs, a way that can be used by other parts of the Globus job-management system and that can be translated into each participating computer's local job-management jargon (for LC, that is the terminology of DPCS and its PSUB utility). The Globus Resource Allocation Manager (GRAM) relies on RSL strings to perform its management functions and to coordinate with other Globus software.

RSL provides

- (1) a syntax used to compose complex resource descriptions from basic elements,
- (2) a way to record resource decisions as attribute-value pairs for later reuse, and
- (3) a vocabulary of job attributes, each of which serves as a parameter to control job behavior when passed to the (remote) system that actually schedules the job.

The Globus web site offers several detailed descriptions of and technical specifications for RSL. This subsection (and the next two) summarize the most important RSL features and the aspects of RSL most relevant for remote users submitting jobs to LC machines.

Start Sentinel:

- & begins all single-job (GRAM-managed) resource descriptions. All Globus jobs destined for LC machines begin their resource specifications this way.
- + begins all multiple-job (DUROC-managed) resource descriptions. Not used for LC-destined jobs (but often in external Globus examples).

Attribute Assignments (Relations):

(attribute = "value")

assigns *value* to the job resource named by *attribute*. Each attribute/value pair is called an RSL relation, and each is separately enclosed in parentheses. A dictionary of allowed RSL attributes appears below (page 19).

Examples:

(executable = "/g/g17/xyz/a+b") (arguments = "a and b")

(attribute = value)

assigns *value* to the job resource named by *attribute*. This simplified syntax without quotes is widely used, and works as long as *value* contains no RSL special characters (see next section (page 18)) or internal white space. Most LC-destined jobs can use this format.

Examples:

(executable = /g/g17/xyz/abc) (maxTime = 1000)

Variable Assignments and Evaluations:

`(rsl_substitution = (VAR "value"))`

assigns *value* to the RSL local variable named *VAR*. You can evaluate these local RSL variables (below) to parameterize a complex RSL file, including even subsequent variable assignments. To assign values to multiple variables, use a space-delimited list of `(VAR "value")` pairs after the equals sign. Variables assigned by using the special `RSL_SUBSTITUTION` attribute apply only to the local RSL file in which they occur; you must invoke a separate `ENVIRONMENT` attribute (with the same syntax) if you intend to assign values to environment variables that apply to your job when it actually runs on the remote host (even same-named variables).

Examples:

`(rsl_substitution = (MYHOST "abc.llnl.gov"))`

`(rsl_substitution = (TOPDIR "/g/g15/jill") (DATADIR "/g/g15/jill/data") (EXECDIR "/g/g15/jill/bin"))`

`(environment (DATADIR "/g/g15/jill/data"))`

`$(VAR)`

evaluates the RSL variable *VAR* and inserts its value here. The order of evaluation is left to right, and outermost to innermost scopes; hence you can even evaluate earlier-assigned variables in the assignment of values to later variables with `RSL_SUBSTITUTION`. Surrounding the expression `$(VAR)` in quotes disables evaluation and turns the expression into a literal (useful for passing it as an argument to your program, for example).

Examples:

`(executable = $(EXECDIR)/myprog)`

`(rsl_substitution (TOPDIR "/g/g15/jill") (DATADIR $(TOPDIR)/data))`

`(arguments = $(DATADIR)/file3 abc "$(MYHOST)")`

Combinations:

`$(VAR) # string`

explicitly concatenates the variable's value with the subsequent string. The RSL explicit concatenation operator (`#`) is seldom needed because ordinary implicit concatenation works in most circumstances.

Examples:

`$(TOPDIR)/bin` and

`$(TOPDIR) # /bin` are equivalent

`(att1 = val1) (att2 = val2) (att3 = val3)`

implicitly conjoins the space-delimited list of attribute/value pairs into a combined resource specification. This implicit conjunction is the typical pattern in most RSL files.

(* comment *) makes a comment. RSL comments are introduced by the prefix (*) and ended by the suffix *). They can occur in any position on a line (where they are treated as blank space), and they cannot be nested.

RSL Special Characters

As with most languages, RSL reserves some characters for special syntactic roles. These special characters cannot appear as part of an unquoted literal; you must either use them for their special role or surround by quotes the literal string in which you want them to appear.

The list of RSL special characters (with roles for the most important) is:

Character	Role
&	GRAM (single-job) start sentinel
+	DUROC (multiple-job) start sentinel
()	delimiters for assignments
\$	variable evaluation
#	explicit concatenation
=	value assignment
" '	mated quotes to protect spaces or special characters within strings
	miscellaneous (all below)
!	
< >	
^	

RSL Attribute Dictionary

Here is an alphabetical explanatory list of the job attributes that the Globus Resource Allocation Manager (GRAM) knows how to recognize, use, and pass along when it finds them in the RSL resource specification for a submitted job. If the attribute has a default value or a corresponding (LC DPCS) PSUB command, they are noted at the end of its description.

`arguments = valuelist`

specifies in order a blank-delimited *valuelist* of command-line arguments to pass to your job's executable. Surround any single value in quotes if a space (or other special RSL character) is required within it.

Example: (argument = "a and b" cde fgh)

Default: NULL

`count = value`

specifies the total number of tasks (executions) across all nodes that your job uses (see also `hostCount`).

Default: 1

PSUB counterpart: -g

`directory = value`

specifies the path for the directory that the (remote, LC) jobmanager will use as the default directory for the submitted job. Using `DIRECTORY` does NOT avoid the need for full pathnames in the job's RSL file, however. Most LC jobs should create (`MKDIR`) their own work directory and move (`CD`) into it as a first step, making use of this attribute unnecessary.

Default: the directory specified by the remote gatekeeper.

`dryRun = yes|no`

toggles between actually running your job (no, the default) and instead verifying its RSL specification (for bad directories, etc.) on the remote machine without running the job (yes). You can override this setting with the `GLOBUSRUN -dryrun execute-line` option.

Default: no

`environment = valuelist`

assigns values to the environment variables that your job will use when it runs remotely, in addition to the default set assigned by the remote jobmanager, where *valuelist* is a blank-delimited list of parenthesis-enclosed variable/value pairs. Note that `ENVIRONMENT` differs from RSL SUBSTITUTION (page 15), which assigns values to variables that apply only within your job's RSL file, not to the job's remote execution environment. `ENVIRONMENT` is equivalent to using `SETENV` (in the C shell) or `var=val;EXPORT` (in the Bourne shell) on the remote machine (which you may do within your job-control script already).

Example: (environment = (DIR /bin) (PI 3.1415))

Default: NULL

PSUB counterpart: see -net, -[no]DFS

executable = *value*

specifies the full pathname of the executable file to run on the remote machine.
Default: none; required attribute

gramMyJob = independent|collective

specifies how the Globus Resource Allocation Manager (GRAM) will behave for this job, where the choices are:

- | | |
|-------------|---|
| independent | starts only one process or thread, even if the COUNT attribute is greater than one. |
| collective | (the default, appropriate for LC) has gram_mjob_count() return a count for each of the processes and gram_myjob_rank() return a unique value between 0 and COUNT-1 for each of the processes. |

hostCount = *value*

specifies the number of nodes your job requires, across which all the tasks in COUNT will be distributed (see also COUNT).

Default: none

PSUB counterpart: -ln

jobType = single|multiple|mpi|condor

specifies how the remote (LC DPCS) jobmanager should start the job, where the choices are:

- | | |
|----------|---|
| single | starts only one process or thread, even if the COUNT attribute is greater than one. |
| multiple | (the default, appropriate for LC) starts COUNT processes or threads. |
| mpi | uses the appropriate method (e.g., POE on an IBM SP) to start a program compiled with a vendor-provided MPI library, using COUNT (not HOSTCOUNT) nodes. |
| condor | starts Condor-library jobs in the Condor universe. |

maxCpuTime = *value*

specifies the maximum CPU time (in minutes) for a single execution of the executable file (must be an integer).

Default: value set by remote scheduler

PSUB counterpart: -lt (see also -tM)

`maxMemory = value`

specifies the maximum amount of memory required for this job (in megabytes).

Default: none

PSUB counterpart: `-c`

`maxTime = value`

specifies the maximum CPU time (in minutes) or wall-clock time (in minutes) for a single execution of the executable file (must be an integer). The GRAM scheduler that handles your job determines whether CPU or wall-clock time is used.

Default: value set by remote scheduler

PSUB counterpart: `-lt` (see also `-tM`)

`maxWallTime = value`

specifies the maximum wall-clock time (in minutes) for a single execution of the executable file (must be an integer).

Default: value set by remote scheduler

PSUB counterpart: `-tW`

`minMemory = value`

specifies the minimum amount of memory required for this job (in megabytes).

Default: none

`project = value`

targets the submitted job to a project (bank, allocation) account as defined by the scheduler on the remote machine.

Default: none

PSUB counterpart: `-b`

`queue = value`

targets the submitted job to a queue (class) name as defined by the scheduler on the remote machine.

Default: none

`stderr = value`

specifies the full pathname of the file to store the standard error messages from the job.

Default: `/dev/null`

PSUB counterpart: `-eo`

`stdin = value`

specifies the full pathname of the file to serve as standard input for the job on the remote machine.

Default: `/dev/null`

`stdout = value`

specifies the full pathname of the file to store the standard output from the job.

Default: `/dev/null`

PSUB counterpart: `-o, -ro`

ASCI Grid Monitoring Services

Applying the Globus approach to (all and only) ASCI resources is how ASCI project managers plan to provide "distributed resource management" (DRM) for ASCI computer users spread among LLNL, LANL, and Sandia National Laboratory (SNL). The basic DRM plan calls for creating an "ASCI Grid" (a trilab-only version of the nationwide "grids" or shared computer networks that Globus makes possible) and managing that special set of shared machines by using established Globus software tools and techniques (layered over the actual local job-control software specific to each lab and its computers).

SNL is leading this ASCI-Grid project, now among open (unclassified) ASCI machines and later among secure (classified) ones. The Globus approach to *monitoring* the ASCI Grid calls for:

- DELIVERING status information through a World-Wide Web interface (a status-report web site) accessed by standard web browsers (see also LLNL's own DISCOM monitoring service at the end of this section), and
- COLLECTING that information by using such existing, underlying Globus services as the Metacomputing Toolkit, the Grid Information Service (GIS), and the LDAP data collectors.

The status-report web site, which exists now but is evolving (access details below), provides three monitoring services for the ASCI Grid, as follows:

STATE-OF-HEALTH.

The state-of-health page on the ASCI Grid Monitoring web site

- Serves primarily *system administrators* who want to check overall DRM status information.
- Reveals the hostname (e.g., `lennier.sandia.gov`) and IP address of the current underlying Globus "data collector" and the time that data were last collected.
- Indicates with a table of colored dots the current status (unknown, active, down) of Globus job-control software on Globus-managed compute resources, including:
 - ◊ The Heartbeat Monitor HBM,
 - ◊ The Globus Resource Allocation Manager GRAM,
 - ◊ The Metacomputing Directory Service MDS.
- Allows optional automatic refresh of this status information once every minute for 10 minutes.

ASCI GRID UTILIZATION.

The grid utilization page on the ASCI Grid Monitoring web site

- Serves primarily *production users* who want to check the status and progress of their own jobs and competing jobs.

- Lists each (Globus-managed) compute resource by host name with a current count of its:
 - ◊ Total CPUs,
 - ◊ Free CPUs,
 - ◊ Active user jobs,
 - ◊ Pending user jobs.
- Graphs the active and pending jobs for all active users, with the option to show jobs for only selected users (picked from a list of user names provided).
- Allows optional automatic refresh of this status information once every minute for 10 minutes.

RESOURCE INFORMATION BASE.

The resource information base (formerly called "browser") page on the ASCI Grid Monitoring web site

- Serves primarily *Globus experts* who know the object names used within Globus databases and who want specific details on specific database entries.
- Starts at the top of the data structure and lets you "drill down" to lower levels by following links provided to every lower-level item.
- Allows display of information by organization, host name, service, queue, and job.
- Displays the attributes and their values for objects at every level in the database, usually in tabular format.

ACCESS.

The open-network version of the ASCI Grid Monitoring Service is located at this unclassified URL (note the https here):

<https://secureweb.sandia.gov/.../dce.sandia.gov/fs/proj/DRM/web/htdocs/monitoring.html>

This web site requires DCE authentication (not compatible with your LC one-time password).

(1) [SNL is still resolving some basic DCE permission issues]

(2) When your browser's authentication window prompts for your user name, supply your *fully qualified* LLNL DCE user name (shown here), not just the last part (*dcename*) normally used to log on to LLNL machines:

`/.../spectrum.llnl.gov/dcename`

(3) When your browser's authentication window prompts for your password, supply the LLNL DCE password that you use to log on to LLNL machines. **WARNING:** If you have fully enabled "DCE credentials" with your LC one-time password (generated by your RSA SecurID authenticator display), then your former DCE password has been disabled and you no longer have any access to the ASCI Grid Monitoring Service (it does *not* accept credentialed OTP authorization).

The secure-network version of the ASCI Grid Monitoring Service is fragmented among several SCF web servers now hosted at LANL:

(1) Start at LLNL's own ICC User Information table at this SCF URL (again, note the https here):

```
https://www.scf.cln/www/sccd
```

(2) Select the "DisCom ASCI Tri-Lab Resources and Services" button from the table. Then on the next page displayed select the "DisCom2 (Distance and Distributed Computing)" button.

(3) This will offer you several different LANL grid-monitoring sites with different roles. The most generally useful is probably the "ASCI Grid Utilization for Los Alamos" button, which corresponds to this SCF URL:

```
http://icnn.lanl.gov/ldswg/drm/cgi-bin/globus-view.cgi
```

WARNING: you will always get an "unable to connect to server" response here unless you execute your web browser on an SCF LLNL ASCI (that is, IBM, *not* Sun or Linux) machine.

DISCOM NETWORK TRAFFIC.

Independently of the official ASCI Grid Monitoring services (above), LLNL maintains a special implementation of NETMON (LC's network-traffic reporting web site) devoted exclusively to network activity among selected DISCOM (distance computing) nodes at LLNL, LANL, and Sandia. The URL for this "DISCOM NETMON" site (on the secure network *only*) is

```
https://lc.llnl.gov/discom
```

(note the "s" in https), and your SCF DCE login and password are required. DISCOM NETMON offers all the usual reports and plots of PING, NETPERF, SSH, DNS, and (four varieties of) FTP network activity among forty pairs of tri-lab nodes. Users unfamiliar with NETMON's customization features should consult the NETMON Reference Manual (URL: <http://www.llnl.gov/LCdocs/netmon>) for suggestions and details.

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government thereof, and shall not be used for advertising or product endorsement purposes.

(C) Copyright 2003 The Regents of the University of California. All rights reserved.

Keyword Index

To see an alphabetical list of keywords for this document, consult the next section (page 27).

Keyword	Description
<u>entire</u>	This entire document.
<u>title</u>	The name of this document.
<u>scope</u>	Topics covered in this document.
<u>availability</u>	Where these programs run.
<u>who</u>	Who to contact for assistance.
<u>introduction</u>	Role and goals of this document.
<u>globus-usage</u>	Using Globus at LC.
<u>globus-inward</u>	Submitting jobs to run at LC.
<u>globus-outward</u>	Submitting outward-bound jobs.
<u>globusrun</u>	How to use the GLOBUSRUN tool.
<u>globusrun-usage</u>	GLOBUSRUN execute line, features.
<u>globusrun-options</u>	Options grouped by role, relevance.
<u>globusrun-examples</u>	Typical LC-relevant samples.
<u>rsl</u>	Resource Specification Language.
<u>rsl-syntax</u>	Syntax summary for LC use.
<u>rsl-characters</u>	RSL special characters noted.
<u>rsl-attributes</u>	Your job's resource needs cataloged.
<u>monitoring-tools</u>	Web monitoring services for ASCI Grid.
<u>asci-grid</u>	Web monitoring services for ASCI Grid.
<u>index</u>	The structural index of keywords.
<u>a</u>	The alphabetical index of keywords.
<u>date</u>	The latest changes to this document.
<u>revisions</u>	The complete revision history.

Alphabetical List of Keywords

Keyword	Description
-----	-----
<u>a</u>	The alphabetical index of keywords.
<u>ascii-grid</u>	Web monitoring services for ASCII Grid.
<u>availability</u>	Where these programs run.
<u>date</u>	The latest changes to this document.
<u>entire</u>	This entire document.
<u>globus-inward</u>	Submitting jobs to run at LC.
<u>globus-outward</u>	Submitting outward-bound jobs.
<u>globus-usage</u>	Using Globus at LC.
<u>globusrun</u>	How to use the GLOBUSRUN tool.
<u>globusrun-examples</u>	Typical LC-relevant samples.
<u>globusrun-options</u>	Options grouped by role, relevance.
<u>globusrun-usage</u>	GLOBUSRUN execute line, features.
<u>index</u>	The structural index of keywords.
<u>introduction</u>	Role and goals of this document.
<u>monitoring-tools</u>	Web monitoring services for ASCII Grid.
<u>revisions</u>	The complete revision history.
<u>rsl</u>	Resource Specification Language.
<u>rsl-attributes</u>	Your job's resource needs cataloged.
<u>rsl-characters</u>	RSL special characters noted.
<u>rsl-syntax</u>	Syntax summary for LC use.
<u>scope</u>	Topics covered in this document.
<u>title</u>	The name of this document.
<u>who</u>	Who to contact for assistance.

Date and Revisions

Revision Date -----	Keyword Affected -----	Description of Change -----
11Jun03	<u>monitoring-tools</u>	DISCOM NETMON site explained.
03Jun02	<u>monitoring-tools</u>	SCF URLs and warnings added.
07Jan02	<u>monitoring-tools</u>	Full OTP disables DCE password and hence all access to this site.
05Oct01	<u>monitoring-tools</u>	OTP access not supported.
07Jun01	<u>monitoring-tools</u> <u>index</u>	New section on monitoring. New keywords for new section.
12Jul00	entire	First edition of LC GLOBUS manual.

TRG (11Jun03)

UCRL-WEB-201501

Privacy and Legal Notice (URL: <http://www.llnl.gov/disclaimer.html>)

TRG (11Jun03) Contact on the OCF: lc-hotline@llnl.gov, on the SCF: lc-hotline@pop.llnl.gov